APPLICATION NO. 09/846,410

TITLE OF INVENTION: Multiple Data Rate Hybrid Walsh Codes for CDMA

INVENTOR: Urbain A. von der Embse

Currently amended CLAIMS

APPLICATION NO. 09/846.410

TITLE OF INVENTION: Multiple Data Rate Hybrid Walsh Codes for CDMA

INVENTORS: Urbain A. von der Embse

CLAIMS

WHAT IS CLAIMED IS:

Claim 1. (cancelled)

Claim 2. (cancelled)

Claim 3. (cancelled)

Claim 4. (cancelled)

Claim 5. (currently amended) A method for generating and applying hybrid Walsh complex orthogonal codes for code division multiple access (CDMA), said method comprising the steps:

generating N Walsh codes W(c) with code index c=0,1,2,. . .,N-1, each with N chips where N is a power of 2,

generating said N hybrid Walsh codes $\tilde{W}(c)$ by reordering each of said N Walsh codes into a corresponding real component and a corresponding imaginary component of a hybrid Walsh code as defined by equations

$$\text{for } c = 0, \quad \tilde{W}(c) = W(0) + jW(0)$$

$$\text{for } c = 1,2,\ldots,N/2-1, \quad \tilde{W}(c) = W(2c) + jW(2c-1)$$

$$\text{for } c = N/2, \quad \tilde{W}(c) = W(N-1) + jW(N-1))$$

$$\text{for } c = N/2+1,\ldots,N-1, \quad \tilde{W}(c) = W(2N-2c-1)+jW(2N-2c)$$

wherein $j=\sqrt{-1}$,

wherein said hybrid Walsh codes are generated by reading the N Walsh codes chip values from a Walsh code memory in a digital signal processor and writing <u>the reordered Walsh codes</u> to a

hybrid Walsh code memory, ~~using addresses specified by said reorderings of said N Walsh codes,~~

applying said hybrid Walsh codes in an encoder and in a decoder of a CDMA system by replacing existing said N Walsh real codes with said hybrid Walsh complex codes according to a same code vector indexing, and

transmitting data encoded by the encoder and receiving data decoded by the decoder.

Claim **6.** (currently amended) A method for generating and applying spreading codes for code division multiple access (CDMA), comprising the steps:

constructing a P by P Discrete Fourier Transform (DFT) matrix E and using DFT as a spreading code with code matrix E wherein row vectors are code vectors and column elements are code elements,

constructing a spreading code from a hybrid Walsh code and ~~a~~ a ~~Discrete Fourier Transform (~~DFT~~)~~ code,

said spreading code is defined by an N*P row by N*P column code matrix C wherein row vectors are code vectors and column elements are code chips,

said hybrid Walsh code is defined by a N row by N column code matrix $\widetilde{W}$,

~~said DFT code is defined by a P row by P column code matrix E,~~

said spreading code matrix C is constructed by a Kronecker product of said hybrid Walsh code matrix $\widetilde{W}$ with said DFT code matrix E defined by the equation

$$C = \widetilde{W} \otimes E$$

wherein the operator "⊗" is a Kronecker product operation,

applying said spreading code matrix C in an encoder and in a decoder of a CDMA system by replacing existing ~~said N~~ real Walsh code

matrix W ~~real codes~~ with said hybrid Walsh complex code~~s~~ matrix C ~~according to a same code vector indexing~~, and

transmitting data encoded by the encoder and receiving data decoded by the decoder.

Claim **7.** (currently amended) A method for implementing hybrid Walsh codes for CDMA, comprising the steps:

encoding N data symbols contained in a block with respective N hybrid Walsh codes to yield N encoded data symbols for each block at the output chip rate of 1/T chips per second wherein T is the interval between chips,

wherein said encoder accepts up to N users per block wherein N is a power of 2 and M is the actual number of users represented in the block, each of said users having a data rate corresponding to one of 1,2,. . . ,N/2 data symbols per block,

wherein said encoder accepts packets from each user and writes them to memory "A" for each block, wherein a binary address index comprising a number of bits corresponding to the maximum number of users N is used for addressing said data symbols stored in memory "A" and the data symbols for each user of the block are stored in memory "A" in a hierarchy such that a particular user is selected according to a number of more significant bits of the binary address index and the data symbols of the particular user are selected according to a number of lesser significant bits of the binary address index, the number of more significant bits and lesser significant bits of the particular user being determined according to the data rate of the particular user and the total number of users M per block.

Claim **8.** (currently amended) Wherein said hybrid Walsh codes in claim **5** have a fast encoding implementation algorithm, comprising the steps:

wherein said fast encoding algorithm implemented in the encoder uses memory "A" for input and to support pass 1 and uses memories "B","C" to support passes 2,. . . ,M wherein N=2^M and uses memory "D" to store the encoded chip output from the reordering pass,

writing input data symbol vector $Z(d_0, d_1,. . .,d_{M-2}, d_{M-1})$ to said "A" wherein the binary addressing word takes address values $d_0d_1{\bullet}{\bullet}{\bullet}{\bullet}d_{M-2}d_{M-1}$=0,1,2,. . .,N-1,

~~wherein pass 1 reads pairs of data symbols from "A" corresponding to d_0=0,1 wherein the addresses of the data symbol for d_0=0 are 0,2,4,. . .,N-2 and for d_0=1 are 1,3,. . .,N-1 and for each pair of data symbols pass 1 performs a 2-point hybrid Walsh transform and sums the outputs for each of the encoded chip binary index values n_{M-1}=0,1 and writes the outputs to memory "B" using the addresses of the respective data symbols corresponding to d_0=0,1, and pass 1 processing generates the data vector $Z(n_{M-1}, d_1,. . .,d_{M-1})$ in "B",~~

wherein

pass m=1~~2,3,. . .,M-1~~ reads pairs of data symbols from "A" and performs a two-point hybrid Walsh transform on the two data symbols in each pair specified by the binary data addresses d_{M-1}=0,1 and writes the output to "B" at the same addresses re-labeled with the binary chip addresses n_0=0,1

pass m-2 reads pairs of data symbols from "B" and performs a two-point hybrid Walsh transform on the two data symbols in each pair specified by the binary data addresses d_{M-2}=0,1 and writes the output to "C" at the same addresses re-labeled with the binary chip addresses n_1=0,1,

pass m=3 continues this processing by reading pairs of data symbols from "C" with the binary addresses d_{M-3}=0,1 and writing the 2-point hybrid Walsh transform output to "B" at the same addresses re-labeled with the binary chip addresses n_2=0,1, ~~"B","C","B",. . . and writes the outputs to "C","B","C",. . .~~

passes m=4,…,M-1 continue this processing using memories "B" and "C",

pass m=M completes the calculation of the fast hybrid Walsh transform by performing a two-point hybrid Walsh transform on the two data symbols specified by the binary data addresses d_0=0,1 and writing the output to the other memory at the same addresses re-labeled with the binary chip addresses n_{M-1}=0,1,

write output of pass m=M is the encoded chip vector $Z(n_{M-1}, . . ., n_0)$ stored in bit-reversed order,

~~and data symbol pairs for d_{m-1}=0,1 are read over address blocks each of length 2^m and starting with the first address block the data symbol d_{m-1}=0 isread for addresses 0,1,,…,2^(m-1) and for d_{m-1}=1 the data symbol addresses are 2^(m-1)+1, . . .,2^m and for each pair of data symbols pass m performs a 2-point hybrid Walsh transform and sums the outputs for each of the encoded chip binary index values (n_{M-m+1} n_{M-m}=0,1)and writes the outputs to addresses (n_{M-m+1},0), (n_{M-m+1},1)corresponding to the addresses of the input d_{m-1}=0,1 and this processing is repeated for each of the address blocks and pass m processing generates the data vector $Z(n_{M-1}, . . ., n_{M-m}, d_m, . . ., d_{M-1})$,~~

~~wherein pass M reads pairs of data symbols from "B" or "C" and writes the outputs to "C" or "B" and data symbol pairs for d_{M-1}=0,1 are read over addresses 0,1,. . .,N/2-1 for d_{M-1}=0 and over addresses N/2,. . .,N-1 for d_{M-1}=1 and for each pair of data symbols pass M performs a 2-point hybrid Walsh transform and sums the outputs for each of the encoded chip binary index values (n_1,n_0=0,1) and writes the outputs to addresses (n_1,0), (n_1,1) corresponding to the addresses of the input d_{M-1}=0,1 and pass M processing generates the data vector $Z(n_{M-1}, . . ., n_0)$,~~

wherein a final reordering pass reorders the encoded chip ~~symbols in memory "B" or "C"~~vector and stores the ordered output chip vector $Z(n_0, n_1, . . ., n_{M-2}, n_{M-1})$ in memory "D", and

wherein said encoder in said CDMA transmitter reads said encoded

chip vector in said "D" and overlays said vector with long and short pseudo-noise (PN) codes to generate N chips of said encoded chip vector for transmission.

Claim **9.** (currently amended) Wherein said hybrid Walsh codes in claim **5** have a fast decoding implementation algorithm, comprising the steps:

wherein the decoder strips off said pseudo-noise (PN) codes from the received N chip encoded chip vector and writes the resultant encoded chip vector $Z(n_0, n_1, . . ., n_{M-2}, n_{M-1})$ to memory "A" wherein the binary addressing word takes address values $n_0 n_1 \bullet\bullet\bullet\bullet n_{M-2} n_{M-1} = 0,1,2,. . ., N-1$,

~~wherein said fast decoding algorithm implemented in said decoder uses said memory "A" for input and to support pass 1 and memories "B","C" support passes 2, . . . ,M and memory "D" stores the decoded data symbols from the rescaling and reordering pass,~~

~~wherein pass 1 reads pairs of encoded chip symbols from "A" corresponding to~~ $\cancel{n_0=0,1}$ ~~wherein the addresses of the chip symbols for~~ $\cancel{n_0=0}$ ~~are 0,2,4, . . .,N-2 and for~~ $\cancel{n_0=1}$ ~~are 1,3, . . .,N-1 and for each pair of chip symbols pass 1 performs a 2-point hybrid Walsh inverse transform and sums the outputs for each of the encoded data symbol index values~~ $\cancel{d_{M-1}=0,1}$ ~~and writes the outputs to memory "B" using the addresses of the respective chip symbols corresponding to~~ $\cancel{n_0=0,1}$ ~~and pass 1 processing generates the vector~~ $\cancel{Z(d_{M-1}, n_1, . . ., n_{M-1})}$ ~~in "B",~~

wherein

pass m=1 reads pairs of chip symbols from "A" and performs a two-point hybrid Walsh inverse transform on the two chip symbols in each pair specified by the binary chip addresses $n_0=0,1$ and writes the output to "B" at the same addresses re-labeled with the binary data addresses $d_{M-1}=0,1$

pass m-2 reads pairs of chip symbols from "B" and performs a

two-point hybrid Walsh inverse transform on the two chip symbols in each pair specified by the binary chip addresses $n_1=0,1$ and writes the output to "C" at the same addresses re-labeled with the binary data addresses $d_{M-2}=0,1$,

pass m=3 continues this processing by reading pairs of chip symbols from "C" with the binary addresses $n_2=0,1$ and writing the 2-point hybrid Walsh inverse transform output to "B" at the same addresses re-labeled with the binary chip addresses $d_{M-3}=0,1$,

passes m=4,…,M-1 continue this processing using memories "B" and "C",

pass m=M completes the calculation of the fast hybrid Walsh inverse transform by performing a two-point hybrid Walsh inverse transform on the two data symbols specified by the binary chip addresses $n_{M-1}=0,1$ and writing the output to the other memory at the same addresses re-labeled with the binary chip addresses $d_0=0,1$,

write output of pass m=M is the data symbol vector $Z(d_{M-1}, \ldots, d_0)$ stored in bit-reversed order, ~~wherein pass m=2,3, . . ., M-1 reads pairs of chip symbols from "B","C","B", . . . and writes the outputs to "C","B","C", . . . and chip symbol pairs for~~ $\cancel{n_{m-1}=0,1}$ ~~are read over address blocks each of length 2^m, and starting with the first address block the chip symbol~~ $\cancel{n_{m-1}=0}$~~is read for addresses 0,1,,…,2^(m-1) and for~~ $\cancel{n_{m-1}=1}$ ~~the chip symbol addresses are 2^(m-1)+1, . . ., 2^m and for each pair of chip symbols pass m performs a 2-point hybrid Walsh inverse transform and sums the outputs for each of the decoded data symbol binary index values~~ $\cancel{(d_{M-m+1}, d_{M-m}=0,1)}$ ~~and writes the outputs to addresses~~ $\cancel{(d_{M-m+1},0), (d_{M-m+1},1)}$ ~~corresponding to the addresses of the input~~ $\cancel{n_{m-1}=0,1}$ ~~andthis processing is repeated for each of the address blocks andpass m processing generates the data vector~~ $\cancel{Z(d_{M-1}, \ldots, d_{M-m}, n_m, \ldots, n_{M-1})}$~~,~~

~~wherein pass M reads pairs of chip symbols from "B" or "C" and writes the outputs to "C" or "B" and chip symbol pairs for~~

~~n_{M-1}=0,1 are read over addresses 0,1,. . .,N/2-1 for n_{M-1}=0 and over addresses N/2,. . .,N-1 for n_{m-1}=1 and for each pair of chip symbols pass M performs a 2-point hybrid Walsh inverse transform and sums the outputs for each of the decoded data symbol binary index values (d_0,d_1=0,1) and writes the outputs to addresses (d_1,0), (d_1,1) corresponding to the addresses of the input n_{M-1}=0,1, and pass M processing generates the data symbol vector Z(d_{M-1}, . . .,d_0),~~

wherein a final pass scales the decoded data symbol~~s symbols~~ vector by the N chip hybrid Walsh inverse transform scaling factor "1/2N" and reorders the scaled ~~encoded~~ data symbol vector ~~s in memory "B" or "C"~~ and stores the ordered data symbol output vector Z(d_0, d_1,. . .,d_{M-2}, d_{M-1}) in memory "D", and

wherein said decoder in said CDMA receiver reads said decoded data symbol vector in said "D" for further processing to recover information from the data symbols.